IN THE CLAIMS

1. (Original) An apparatus, in a data processing system having at least one host processor with host processor cache and host memory, the apparatus comprising:

   a chip interconnect;

   a cache coherent interface coupled to the chip interconnect, the cache coherent interface providing cache coherent access;

   a cache non-coherent interface coupled to the chip interconnect, the cache non-coherent interface providing cache non-coherent access; and

   a compute engine coupled to the chip interconnect, coupled to the cache coherent interface, and coupled to the cache non-coherent interface, the compute engine issuing a memory access request.

2. (Original) The apparatus of claim 1, wherein the cache coherent interface receives a cache coherent access request from the compute engine, and wherein the cache non-coherent interface receives a cache non-coherent access request from the compute engine.

3. (Original) The apparatus of claim 1, further comprising multiple compute engines, the multiple compute engines issuing memory access requests simultaneously.

4. (Original) The apparatus of claim 3, wherein the cache coherent interface and cache non-coherent interface are capable of handling multiple memory access requests from the multiple compute engines simultaneously.

5. (Original) The apparatus of claim 1, wherein if the memory access request is targeted for cache coherent access, the cache coherent interface transmits a cache coherency message to the host processor.

6. (Original) The apparatus of claim 5, wherein if the memory access request is a cache coherent read access, the host processor flushes the host processor cache to the host memory if the host processor cache contains newer data, and thereafter, the cache coherent interface reads data from the host processor cache or from the host memory.

7. (Original) The apparatus of claim 5, wherein if the memory access request is a cache coherent write access, the cache coherent interface writes data to the host memory, and thereafter, the host processor refreshes the host processor cache from the host memory.

8. (Original) The apparatus of claim 1, wherein if the memory access request is targeted for cache coherent access, the cache coherent interface examines the memory access request to determine if there is a cache coherency violation.

9. (Original) The apparatus of claim 8, wherein if there is a cache coherency violation, the cache coherent interface issues an error message through an interrupt mechanism.

10. (Original) The apparatus of claim 1, wherein if the memory access request is targeted for cache non-coherent access, the cache non-coherent interface provides direct accesses to the host memory.

11. (Original) The apparatus of claim 1, further comprising:

a host interface coupled to the chip interconnect, the host interface providing access to the host processor;

a memory interface coupled to the chip interconnect, the memory interface providing access to the host memory; and

a PCI-0 interface coupled to the chip interconnect.

12. (Original) The apparatus of claim 11, wherein if the memory access request is targeted for cache coherent access, the host interface receives a cache coherency message from the cache coherent interface, and transmits the cache coherency message to the host processor.

13. (Original) The apparatus of claim 12, wherein if the memory access request is a cache coherent read access, the host processor flushes the host processor cache to the host memory, and thereafter, the cache coherent interface reads data from the host processor cache or the host memory, through the memory interface.

14. (Original) The apparatus of claim 12, wherein if the memory access request is a cache coherent write access, the cache coherent interface writes data to the host memory, and thereafter, the host processor refreshes the host processor cache from the host memory.

15. (Original) The apparatus of claim 12, wherein if the memory access request is targeted for cache coherent access, the host interface receives a cache coherency message from the cache coherent interface and examines the memory access request to determine if there is a cache coherency violation.

16. (Original) The apparatus of claim 15, wherein if there is a cache coherency violation, the host interface issues an error message through an interrupt mechanism.

17. (Original) The apparatus of claim 11, wherein the PCI-0 interface is an accelerated graphics port (AGP), the AGP being configured to coupled to an external graphics controller.

18. (Original) The apparatus of claim 1, further comprising a video address re-mapping table (VART) coupled to the cache coherent interface, coupled to the cache non-coherent

interface, and coupled to the compute engine, the VART performing translation of a logical address to a physical address.

19. (Original) The apparatus of claim 18, wherein the VART determines if the physical address is targeted for cache coherent access, wherein the VART transmits the memory access request to the cache coherent interface if the physical address is targeted for cache coherent access, and wherein the VART transmits the memory access request to the cache non-coherent interface if the physical address is targeted for cache non-coherent access.

20. (Original) The apparatus of claim 18, wherein the compute engine determines if the memory access request needs to be mapped, wherein the compute engine transmits the memory access request to the VART if the memory access request needs to be mapped.

21. (Original) The apparatus of claim 20, wherein if the memory access request does not need to be mapped, the compute engine transmits the memory access request for cache coherent access to the cache coherent interface and the compute engine transmits the memory access request for cache non-coherent access to the cache non-coherent interface.

22. (Original) The apparatus of claim 1, further comprising a memory management controller coupled to the cache coherent interface and the cache non-coherent interface, the memory management controller being coupled to the compute engine.

23. (Original) The apparatus of claim 22, wherein the memory management controller receives the memory access request from the compute engine and determines if the memory access request is targeted for cache coherent access, wherein the memory management controller transmits the memory access request to cache coherent interface

if the memory access request is targeted for cache coherent access, and wherein the memory management controller transmits the memory access request to cache non-coherent interface if the memory access request is targeted for cache non-coherent access.

24. (Original) The apparatus of claim 22, wherein the memory management controller receives the memory access request and determines if the memory access request needs to be mapped, wherein the memory management controller transmits the memory access request to a video address re-mapping table (VART) if the memory access request needs to be mapped, and wherein the memory management controller transmits the memory access request to the cache coherent interface or cache non-coherent interface if the memory access request does not need to be mapped.

25. (Original) A method in a computer system having at least one host processor with host processor cache and host memory coupled to a chipset having a cache coherent interface and a cache non-coherent interface, the method comprising:

receiving a memory access request from a compute engine;

examining the memory access request to determine whether the memory access request is targeted for cache coherent access;

wherein if the memory access request is targeted for cache coherent access, transmitting the memory access request to the cache coherent interface; and

wherein if the memory access request is targeted for cache non-coherent access, transmitting the memory access request to the cache non-coherent interface.

26. (Original) The method of claim 25, further comprising multiple compute engines issuing multiple memory access requests simultaneously.

27. (Original) The method of claim 25, further comprising transmitting a cache coherency message to the host processor from the cache coherent interface, if the memory access request is targeted for cache coherent access.

28. (Original) The method of claim 27, wherein if the memory access request is a cache coherent read access, the method further comprises:

flushing the host processor cache to the host memory if the host processor cache contains newer data; and

reading data from the host processor cache or the host memory through the cache coherent interface, after the flushing.

29. (Original) The method of claim 27, wherein if the memory access request is a cache coherent write access, the method further comprises:

writing data to the host memory through the cache coherent interface; and

refreshing the host processor cache from the host memory, after the writing.

30. (Original) The method of claim 25, wherein if the memory access request is targeted for cache coherent access, the method further comprises examining the memory access request by the cache coherent interface, to determine if there is a cache coherency violation.

31. (Original) The method of claim 30, wherein if there is a cache coherency violation, the method further comprises issuing an error message through an interrupt mechanism.

32. (Original) The method of claim 25, wherein if the memory access request is targeted for cache non-coherent access, the method further comprises accessing the host memory directly through the cache non-coherent interface.

33. (Original) The method of claim 25, further comprising:

examining the memory access request to determine if the memory access request requires to be mapped from a logical address to a physical address;

translating the logical address to the physical address through a memory mapping mechanism, if the memory access request requires to be mapped;

examining the physical address to determine if the memory access request is configured for cache coherent access;

transmitting the memory access request to the cache coherent interface if the memory access request is configured for cache coherent access; and

transmitting the memory access request to the cache non-coherent interface if the memory access request is configured for cache non-coherent access.

34. (Original) The method of claim 33, wherein the memory mapping mechanism is a video address re-mapping table (VART).

35. (Original) The method of claim 33, wherein the memory mapping mechanism examines the physical address to determine if the memory access request is targeted for cache coherent access.

36. (Original) The method of claim 33, wherein the memory mapping mechanism examines the physical address to determine if the physical address is valid and wherein if the physical address is invalid, the method further comprises generating an error message through an interrupt mechanism.

37. (Original) The method of claim 33, further comprising:

transmitting the memory access request to the memory mapping mechanism if the memory access request requires to be mapped; and transmitting the memory access request to the cache coherent interface or the cache non-coherent interface if the memory access request does not require to be mapped.

38. (Original) The method of claim 25, wherein the chipset further comprises:

at least one compute engine issuing the memory access request;

a host interface coupled to a host processor;

a memory interface coupled to a host memory; and

a PCI-0 interface.

39. (Original) The method of claim 38, wherein the PCI-0 interface is an accelerated graphics port (AGP), the AGP being configured to coupled to an external graphics controller.

40. (Original) The method of claim 38, further comprising:

examining the memory access request to determine the destination of the memory access request;

transmitting the memory access request to the memory interface if the memory access request is targeted for the host memory;

transmitting the memory access request to a compute engine if the memory access request is targeted for the compute engine; and

transmitting the memory access request to a PCI-0 interface if the memory access request is targeted for the PCI-0 interface.

41. (Original) The method of claim 40, wherein if the memory access request is targeted for the compute engine, the method further comprises determining if the compute engine is valid by examining an ID number of the compute engine.

42. (Original) The method of claim 40, wherein if the memory access request is for PCI-0 interface, the method further comprises determining if the memory access request is valid using a PCI-0 address mask register located in the PCI-0 interface.

43. (Original) An apparatus, having a cache coherent interface and a cache non-coherent interface, in a computer system having at least one host processor with host processor cache and host memory, the apparatus comprising:

means for receiving a memory access request from a compute engine;

means for examining the memory access request to determine whether the memory access request is targeted for cache coherent access;

wherein if the memory access request is targeted for cache coherent access, means for transmitting the memory access request to the cache coherent interface; and

wherein if the memory access request is targeted for cache non-coherent access, means for transmitting the memory access request to the cache non-coherent interface.

44. (Original) The apparatus of claim 43, further comprising multiple compute engines issuing multiple memory access requests simultaneously.

45. (Original) The apparatus of claim 43, further comprising means for transmitting a cache coherency message to the host processor from the cache coherent interface, if the memory access request is targeted for cache coherent access.

46. (Original) The apparatus of claim 45, wherein if the memory access request is a cache coherent read access, the apparatus further comprises:

means for flushing the host processor cache to the host memory if the host processor cache contains newer data; and

means for reading data from the host processor cache or the host memory through the cache coherent interface, after the flushing.

47. (Original) The apparatus of claim 45, wherein if the memory access request is a cache coherent write access, the apparatus further comprises:

means for writing data to the host memory through the cache coherent interface; and

means for refreshing the host processor cache from the host memory, after the writing.

48. (Original) The apparatus of claim 43, wherein if the memory access request is targeted for cache coherent access, the apparatus further comprises means for examining the memory access request by the cache coherent interface, to determine if there is a cache coherency violation.

49. (Original) The apparatus of claim 48, wherein if there is a cache coherency violation, the apparatus further comprises means for issuing an error message through an interrupt mechanism.

50. (Original) The apparatus of claim 43, wherein if the memory access request is targeted for cache non-coherent access, the apparatus further comprises means for accessing the host memory directly through the cache non-coherent interface.

51. (Original) The apparatus of claim 43, further comprising:

means for examining the memory access request to determine if the memory access request requires to be mapped from a logical address to a physical address;

means for translating the logical address to the physical address through a memory mapping mechanism, if the memory access request requires to be mapped;

means for examining the physical address to determine if the memory access request is configured for cache coherent access;

means for transmitting the memory access request to the cache coherent interface if the memory access request is configured for cache coherent access; and means for transmitting the memory access request to the cache non-coherent interface if the memory access request is configured for cache non-coherent access.

52. (Original) The apparatus of claim 51, wherein the memory mapping mechanism is a video address re-mapping table (VART).

53. (Original) The apparatus of claim 51, wherein the memory mapping mechanism examines the physical address to determine if the memory access request is targeted for cache coherent access.

54. (Original) The apparatus of claim 51, wherein the memory mapping mechanism examines the physical address to determine if the physical address is valid and wherein if the physical address is invalid, the apparatus further comprises means for generating an error message through an interrupt mechanism.

55. (Original) The apparatus of claim 51, further comprising:

means for transmitting the memory access request to the memory mapping mechanism if the memory access request requires to be mapped; and

means for transmitting the memory access request to the cache coherent interface or the cache non-coherent interface if the memory access request does not require to be mapped.

56. (Original) The apparatus of claim 43, further comprising:

at least one compute engine issuing the memory access request;

a host interface coupled to a host processor;

a memory interface coupled to a host memory; and

a PCI-0 interface.

57. (Original) The apparatus of claim 56, wherein the PCI-0 interface is an accelerated graphics port (AGP), the AGP being configured to coupled to an external graphics controller.

58. (Original) The apparatus of claim 56, further comprising:

means for examining the memory access request to determine the destination of the memory access request;

means for transmitting the memory access request to the memory interface if the memory access request is targeted for the host memory;

means for transmitting the memory access request to a compute engine if the memory access request is targeted for the compute engine; and

means for transmitting the memory access request to a PCI-0 interface if the memory access request is targeted for the PCI-0 interface.

59. (Original) The apparatus of claim 58, wherein if the memory access request is targeted for the compute engine, the apparatus further comprises means for determining if the compute engine is valid by examining an ID number of the compute engine.

60. (Original) The apparatus of claim 58, wherein if the memory access request is for PCI-0 interface, the apparatus further comprises means for determining if the memory access request is valid using a PCI-0 address mask register located in the PCI-0 interface.

61. (Original) A machine readable medium having stored thereon executable code which causes a machine to perform a method, in a computer system having at least one host processor with host processor cache and host memory coupled to a chipset having a cache coherent interface and a cache non-coherent interface, the method comprising:

receiving a memory access request from a compute engine;

examining the memory access request to determine whether the memory access request is targeted for cache coherent access;

wherein if the memory access request is targeted for cache coherent access, transmitting the memory access request to the cache coherent interface; and

wherein if the memory access request is targeted for cache non-coherent access, transmitting the memory access request to the cache non-coherent interface.

62. (Original) The machine readable medium of claim 61, wherein the method further comprises multiple compute engines issuing multiple memory access requests simultaneously.

63. (Original) The machine readable medium of claim 61, wherein the method further comprises transmitting a cache coherency message to the host processor from the cache coherent interface, if the memory access request is targeted for cache coherent access.

64. (Original) The machine readable medium of claim 63, wherein if the memory access request is a cache coherent read access, the method further comprises:

flushing the host processor cache to the host memory if the host processor cache contains newer data; and

reading data from the host processor cache or the host memory through the cache coherent interface, after the flushing.

65. (Original) The machine readable medium of claim 63, wherein if the memory access request is a cache coherent write access, the method further comprises:

writing data to the host memory through the cache coherent interface; and

refreshing the host processor cache from the host memory, after the writing.

66. (Original) The machine readable medium of claim 61, wherein if the memory access request is targeted for cache coherent access, the method further comprises examining the memory access request by the cache coherent interface, to determine if there is a cache coherency violation.

67. (Original) The machine readable medium of claim 66, wherein if there is a cache coherency violation, the method further comprises issuing an error message through an interrupt mechanism.

68. (Original) The machine readable medium of claim 61, wherein if the memory access request is targeted for cache non-coherent access, the method further comprises accessing the host memory directly through the cache non-coherent interface.

69. (Original) The machine readable medium of claim 61, wherein the method further comprises:

examining the memory access request to determine if the memory access request requires to be mapped from a logical address to a physical address;

translating the logical address to the physical address through a memory mapping mechanism, if the memory access request requires to be mapped;

examining the physical address to determine if the memory access request is configured for cache coherent access;

transmitting the memory access request to the cache coherent interface if the memory access request is configured for cache coherent access; and

transmitting the memory access request to the cache non-coherent interface if the memory access request is configured for cache non-coherent access.

70. (Original) The machine readable medium of claim 69, wherein the memory mapping mechanism is a video address re-mapping table (VART).

71. (Original) The machine readable medium of claim 69, wherein the memory mapping mechanism examines the physical address to determine if the memory access request is targeted for cache coherent access.

72. (Original) The machine readable medium of claim 69, wherein the memory mapping mechanism examines the physical address to determine if the physical address is valid and wherein if the physical address is invalid, the method further comprises generating an error message through an interrupt mechanism.

73. (Original) The machine readable medium of claim 69, wherein the method further comprises:

transmitting the memory access request to the memory mapping mechanism if the memory access request requires to be mapped; and

transmitting the memory access request to the cache coherent interface or the cache non-coherent interface if the memory access request does not require to be mapped.

74. (Original) The machine readable medium of claim 61, wherein the chipset further comprises:

at least one compute engine issuing the memory access request;

a host interface coupled to a host processor;

a memory interface coupled to a host memory; and

a PCI-0 interface.

75. (Original) The machine readable medium of claim 74, wherein the PCI-0 interface is an accelerated graphics port (AGP), the AGP being configured to coupled to an external graphics controller.

76. (Original) The machine readable medium of claim 74, wherein the method further comprises:

examining the memory access request to determine the destination of the memory access request;

transmitting the memory access request to the memory interface if the memory access request is targeted for the host memory;

transmitting the memory access request to a compute engine if the memory access request is targeted for the compute engine; and

transmitting the memory access request to a PCI-0 interface if the memory access request is targeted for the PCI-0 interface.

77. (Original) The machine readable medium of claim 76, wherein if the memory access request is targeted for the compute engine, the method further comprises determining if the compute engine is valid by examining an ID number of the compute engine.

78. (Original) The machine readable medium of claim 76, wherein if the memory access request is for PCI-0 interface, the method further comprises determining if the memory access request is valid using a PCI-0 address mask register located in the PCI-0 interface.